

Introduction to Task Flow Diagrams

Pre-work 2 ***(To be completed after Pre-work 1)***



December 2005

What is Task Flow Diagramming?

What is the purpose of task flow diagramming?

In context analysis, we map out the *environment* in which tasks occur, showing all the entities in a business process and the transactions among them (see pre-work reading 1). However, when creating context diagrams, we do not consider the activities that occur *within* each entity. Task flow diagrams are a tool for us to portray these activities, or tasks.

Task flow diagrams proceed from, and are more detailed than, context diagrams. Key tasks—those that are most important to identified requirements—are described in more detail in task flow diagrams.

Mapping out important tasks gives us more detail about how the information systems can support activity within a business process.

What form do task flow diagrams take?

Task flow diagrams identify the various functional tasks required to achieve the organizational objective derived from the Context Analysis. The diagrams also show the way in which tasks relate to one another since results from one task are used as inputs to others. Each functional task conforms to the basic data processing model of input, process, and output (result) as depicted in figure 1.

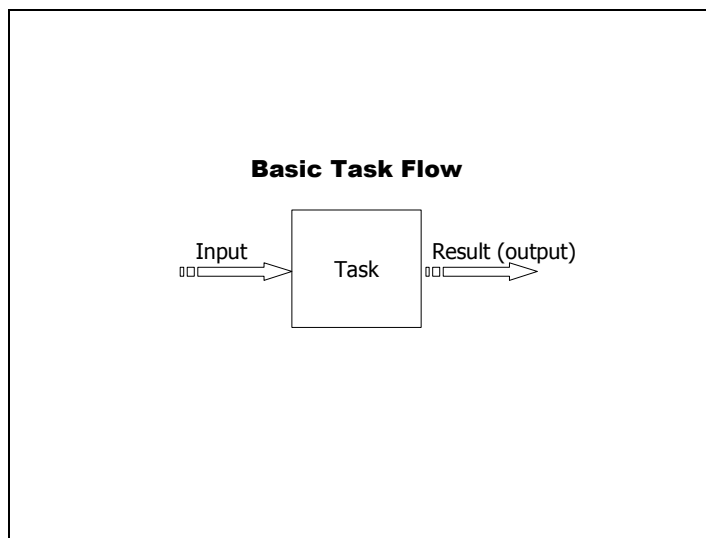
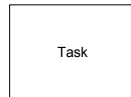


Figure 1

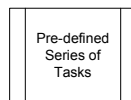
Task flow diagrams look like standard flowcharts. Therefore, they have the following qualities:

- There is an input, or starting point.
- There are one or more activities or tasks performed on the input.
- There is an output.
- They are read from left to right or top to bottom. Task flow diagrams are networks with a single direction of flow, not tree-structured hierarchies.
- Standard flowcharting tools are used—rectangles, arrows, diamonds for decision points, etc.

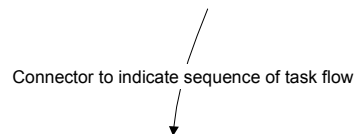
Task Flow Symbols



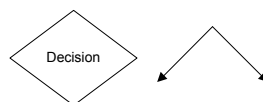
A 'task box' depicts an activity or series of activities that can be performed by an individual or group from beginning to end without interruption (from another entity or task).



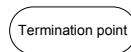
A box with parallel lines along each side is a shorthand symbol to represent a series of tasks that is grouped together to execute a specific operation.



Lines with arrows are drawn to connect symbols and to indicate direction of task sequencing. Task flows are read from top to bottom or from left to right.



A decision 'box' (really a diamond or rotated box) shows an "either/or" condition is being invoked by the entity performing the work. Depending upon the answer to the question posed within the diamond, one of two or more courses of action will be followed. Sometimes the actual box is implied with the use of split arrows.



A rectangle with rounded corners is the symbol used to identify the beginning and end to any series of tasks.



The on and off-page connectors are used to show that a sequence of tasks is broken (usually due to space on a page) and continues somewhere else. The letters or numbers or text that appears inside the connectors are matched to the mating connector where the sequence is continued.



We can convert context diagrams, or parts of them, into task flow diagrams simply by changing the graphical look of them. For example, in prework reading 1, we constructed the following context diagram:

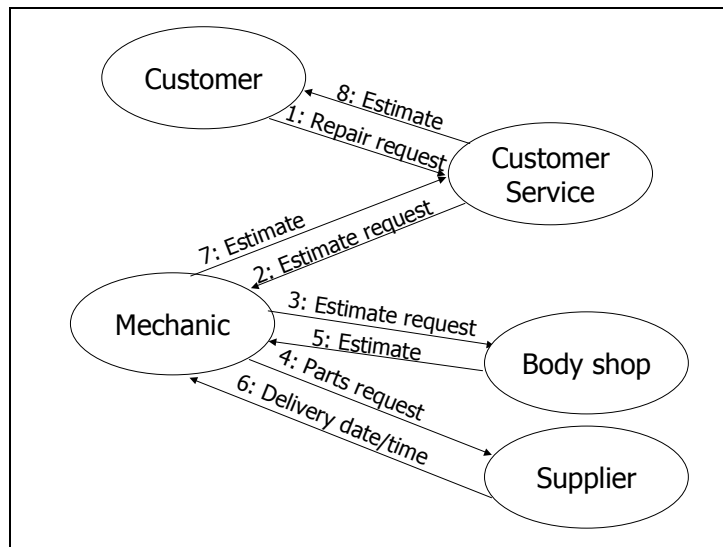


Figure 2

If we wanted to convert this context diagram to a task flow diagram, it would look like this:

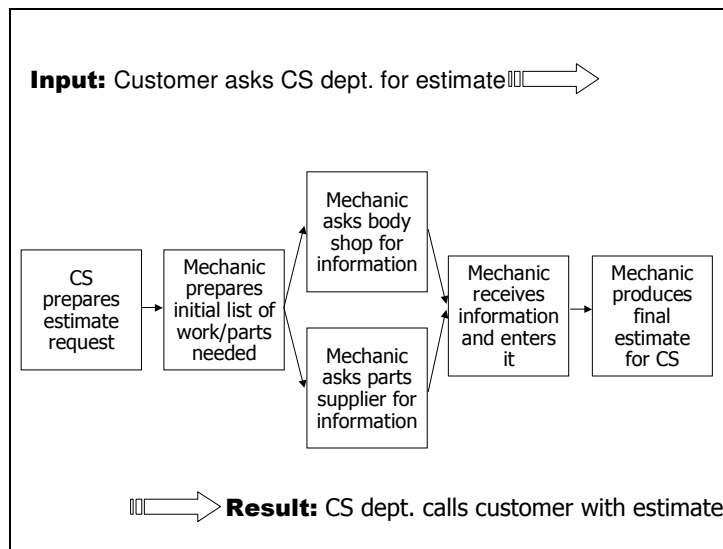


Figure 3

When do we complete task flow diagrams?

If task flow diagrams were just context diagrams in another form, there would be no need for them. However, context diagrams and task flow diagrams serve two different purposes:

- Context diagrams give us an overall structure of the entire environment in which a process takes place. They help us make sure that we are identifying all entities and transactions between entities—key transfer points where information systems need to convey information.
- Task flow diagrams help us break down more specifically the discrete tasks that occur within a particular entity. Discrete tasks are those which can be performed by an individual or group without interruption once the task has started.

Task flow diagrams are completed for key areas of the process description (where it is important to look at the detailed, discrete task level).

For example, in the box in figure 3 that reads, “Mechanic produces final estimate for CS,” there are several tasks that take place *within* this box that need to be mapped out. Converting this to a simple task flow diagram (input/process/output) would look like this:

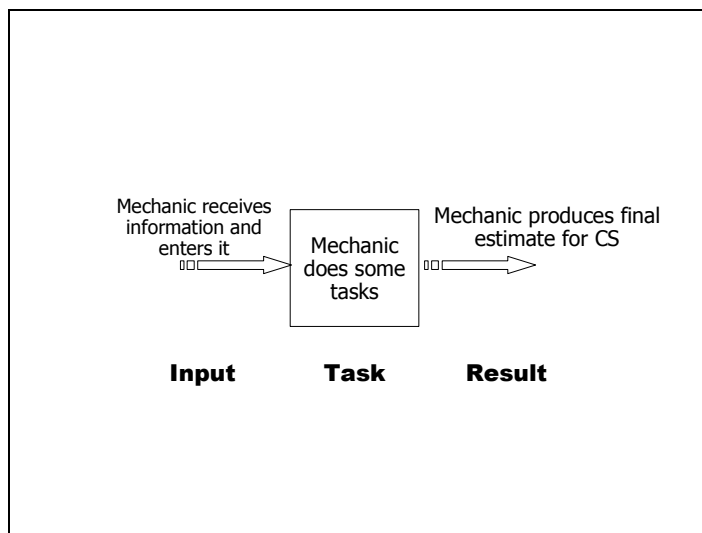


Figure 4

Now, the business analyst asks a series of questions to determine what those tasks are. The result is a more specific task flow diagram that looks like this:

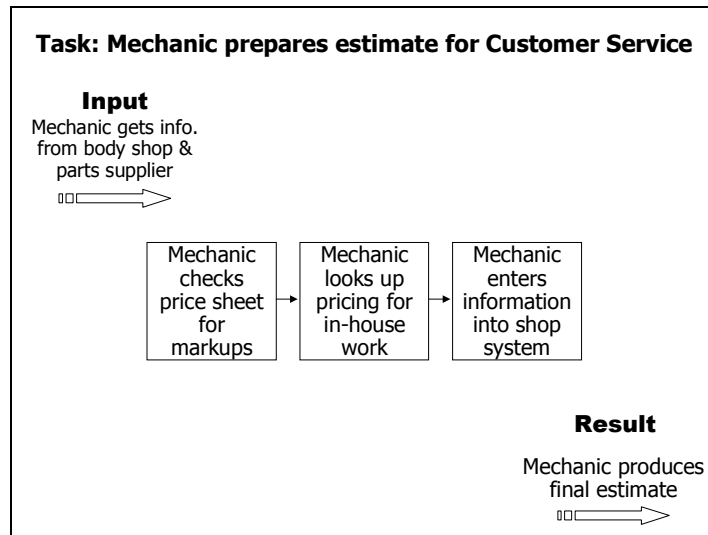


Figure 5

Other uses for task flow diagrams:

- Error handling and exceptions. If the detail level diagram doesn't include all exception or error handling processes within the environment being analyzed, additional detail may be needed to adequately show the exception/ error handling functional flows. These diagrams should be constructed at the detail diagram level using discrete functional flow tasks. Entity diagrams may be built to assist in development of the flow diagram if desired. The alternate flows may be shown separately or integrated into the primary detail functional flow diagram.
- Improved task flow diagrams. These detail level diagrams are used to show the functional flow steps and their relationships as they will be performed when the new system support is instituted.

How do we create task flow diagrams?

Start with the Context Diagram

To create task flow diagrams, we start by creating a flowchart that is basically a repeat of the information in the context diagram, except in format—the format should be that of a flowchart, as we did when converting the context diagram to an *overall* task flow diagram.

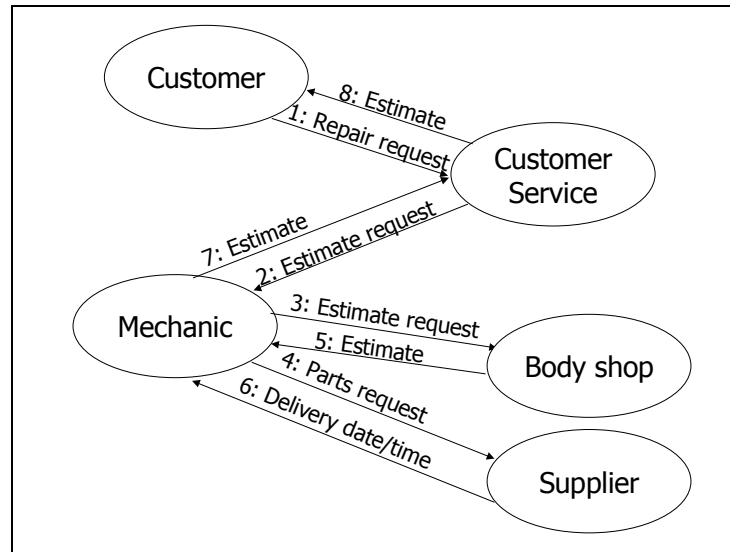


Figure 6: Context Diagram

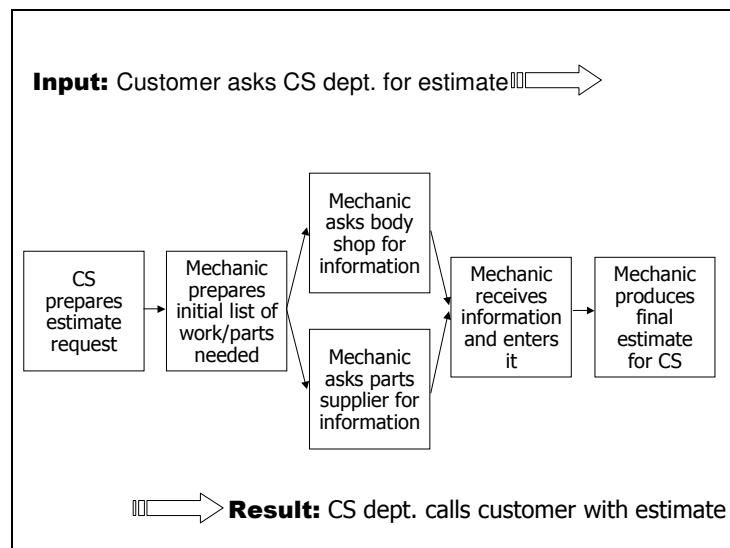


Figure 7: Overall Task Flow Diagram

The flow is always from left to right or top to bottom and the diagram as a whole represents a complete, single pass through the functional environment described.

Build task flows by looking at each part of the overall task flow diagram

Once you have the overall task flow diagram (above) , look at each step in the flow to determine whether it represents a single, discrete activity or multiple activities performed by an internal entity

How do we identify results?

Results from one step are used as inputs into the next sequential task and, in some cases, other downstream functional tasks. Frequently, one subset of the results is used in the next step and another subset used elsewhere. Results may also directly represent the achievement of the organizational objective being analyzed, that is, no additional functional task is required to achieve the objective. Any result directly meeting an end objective can be viewed as "the last link in the chain."

Thus, the result from each functional task is used in three distinct ways. First, as input to the next sequential step. Second, as input to a downstream step. Third, to represent the achievement of an organizational objective.

For example, the last box in the diagram below is, "Mechanic produces final estimate for CS."

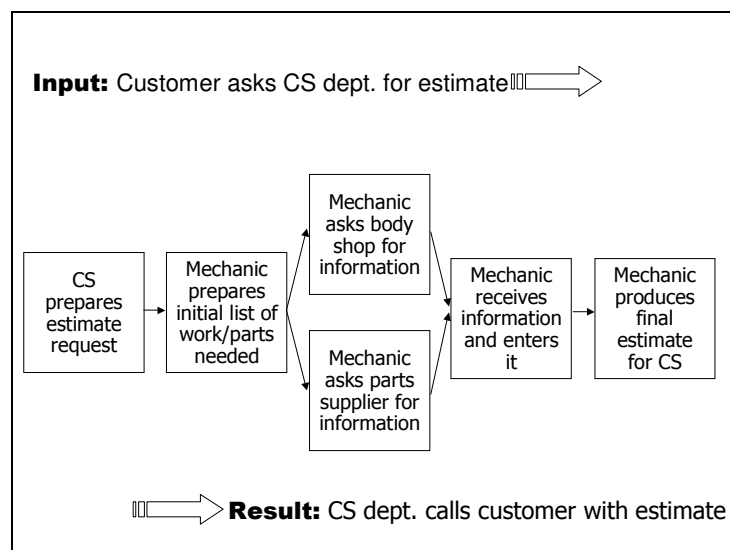


Figure 8

The completion of the estimate serves as both a result of the task within the mechanic entity AND an input for the customer service department (next box in task flow).

Naming task flow diagrams. Usually, the name of a task usually includes the entity and the result of the task. For example, the task ending with the information being sent to the outside entities (body shop and parts supplier) might be titled, "Mechanic asks body shop and supplier for information."

A note about cyclical processes: Since most data processing systems support cyclical functional environments, handling of the repeating cycles can be a stumbling block. The basic rule, however, is simple: Stop once the steps become redundant. Identify the achievement of the objective and any results from the cycle required as inputs to the next cycle. At the point where they are used as inputs, note that they are from the previous cycle. For example, when producing a monthly bank statement, the input is simply the balance from the previous cycle, and should be shown as such.

How do we identify inputs?

Inputs are derived from these sources:

- They may be transactions on the entity diagram being received by the entity performing the task. That is, all transactions with the heads of the arrows pointing at the given entity are candidates for use as inputs.
- They can be the result from a prior process.
- They can be data from a previous cycle.
- They can be "raw" input into the process from outside the process. Raw inputs are all inputs not covered by the other two categories.

Therefore, each box on our overall task flow diagram, except for the last one, is an input for a task within an entity.

In the example we talked about above, “Mechanic prepares estimate for Customer Service,” the input would be the result of the previous step, “body shop and supplier give information.”

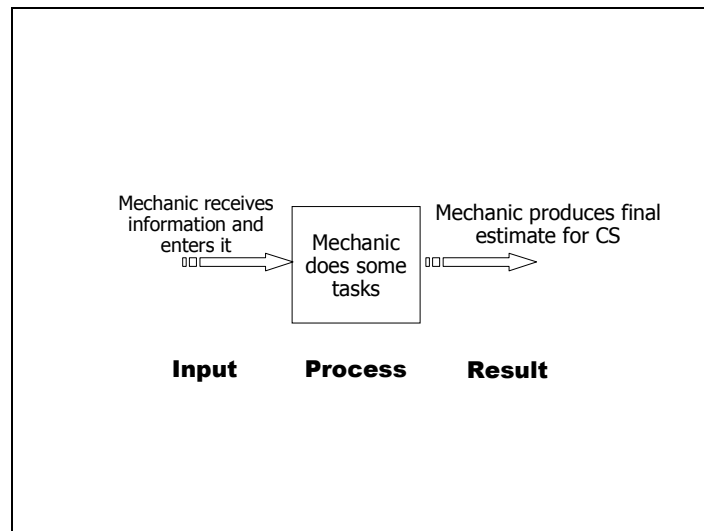


Figure 9

How do we identify transactions?

In addition to inputs and results, the diagram must also describe the actions involved to complete the basic input-process-output task description.

Upon further examination of the mechanic’s work associated with finalizing the estimate, we discover that several tasks must be performed. First, the mechanic must check price sheets to see what the markup is on parts and auto body work. Then, he must look up pricing for work that will be completed in-house at the repair shop. He then enters this information into an estimator sheet in the shop system, which totals the information in a format for him to print out and give to customer service.

Thus the single transaction is shown as the following task flow:

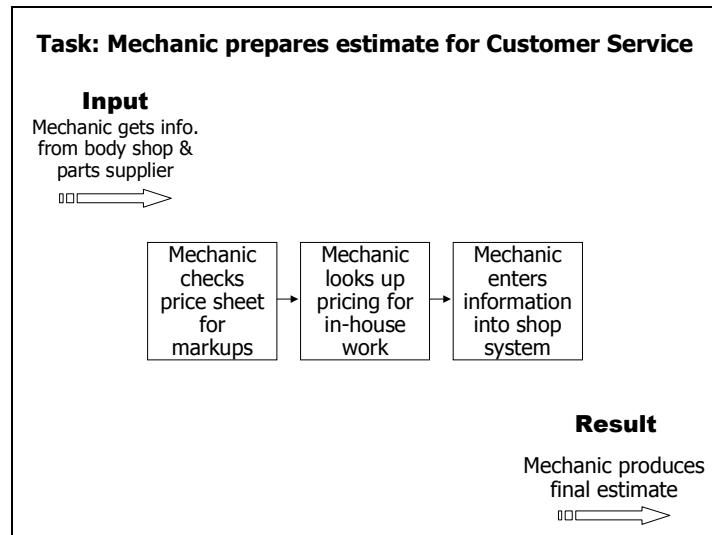


Figure 10

Completing a series of task flow diagrams

One context diagram will produce multiple task flow diagrams. By working from one transaction to another, and expanding those that are most relevant to the project goals and scope, we gradually build a series of individual task flow diagrams that collectively describe the process.

Think carefully about how to define discrete tasks. Discrete tasks should only be delineated if they are performed by an entity under the organization's control. External entities, by definition, are not under the sponsoring organization's control. Therefore it is unlikely that the contemplated system support will support the performance of tasks undertaken by any external entity. It then follows that the overall task flow description is sufficient for all activities performed by external entities.

For example, we would NOT develop a task flow diagram for "body shop and supplier give information" (fourth step in figure 8)," because we are not concerned about the processes that the body shop and parts supplier (external entities) go through to get the information, only the information that comes back to us.

Even if a functional activity is performed by an internal entity it is necessary to break it down further only if the activity defines multiple, unique processes that should be broken out separately. The rule most helpful in this situation is to evaluate the activity in terms of whether or not it represents a discrete work activity that requires a set of inputs from which a set of results intended to be passed to another person, or group, are produced. Visualize it as a desk with an in-basket and an out-basket. As long as the work involves using the contents of the in-basket to create a result that ends up in the out-basket, it is a single functional activity (unit of work) no matter how many individually identifiable work steps are required to produce the result. If the person performing the work prepared a work paper is used internally to perform a subsequent step, the entire sequence should be treated as a single functional activity.

The key words in the determination of discrete tasks are *continuous* and *uninterrupted*. If the task can be performed by an individual or group from beginning to end without interruption (continuous from beginning to end) then it is a discrete work activity.

One transaction may result in multiple task flow diagrams.

Discrete tasks performed within a given entity would not show up on the overall task map because it would require a subdivision of the natural entity. Thus the "Order Entry Department" would have to appear as "Jane's Desk", "Sally's Desk", and "Joe's Desk" if there were three discrete processes performed on order entry. For example if Jane took the orders over the phone, Sally verified the product codes and price, and Joe formally approved the order then there would be three discrete tasks performed in sequence within the order entry department. To reflect these transactions on the Context diagram would require the creation of artificial entities; the distinction should be spelled out here, during task flow diagramming.

Keeping track of the work. Whenever a transaction from the entity diagram is used as an input into a process, it should be "crossed off" the entity diagram, although it can be used multiple times. At the end of the mapping process, all transactions should appear at least once on the functional flow diagram. Thus all transactions should be "crossed off" at the end of the mapping process.

NOTE: Ultimately, any separate diagrams should tie together when the analysis is completed. That is, there should be only one overall purpose or objective in the final result, with all other objectives mapping into (contributing to) the achievement of the overall final purpose or objective. If the analysis results in multiple diagrams it indicates that the entity diagram produced a result of the context analysis has described two or more total independent business processes.

How do we use task flow diagrams to make improvements?

First, establish “what is”

The initial pass through the functional environment establishes the definition of how the functionality is performed under the current level of systems support and operational procedures.

The introduction of new support through the installation of the system being designed provides an opportunity to upgrade and streamline the functional activities to maximize the impact of the system support.

Sometime the question arises regarding the need to analyze and document the existing functional environment if you are going to change it. The answer is that unless you can establish a baseline (the existing environment) you will never be able to measure the benefit to be derived by introducing change. Further, many users themselves do not understand the way in which the organization functions. They may have a narrow view of their own operation, and maybe a feel for the operations providing them input and those to which they send results. Rarely will they have a good feel for the whole fabric.

Also, without a systematic analysis of the current environment it is very difficult to introduce change. Users will not vote for change if they don't completely understand the impact on the organization and the way in which the current business processes achieve business objectives. They need a high comfort level, particularly if the functional procedures and tasks have been in place a long time.

An opportunity to improve

One of the greatest opportunities for benefit to the organization is in the area of improving the way in which a business purpose or objective is achieved in the functional environment.

Typically, any thought given to improving the functional flow is generally limited to narrow departmental boundaries. Thus the conceptual framework within which alternatives are considered excludes thinking about the overall business process as a whole, which normally cuts across multiple organizational boundaries.

The participants must adopt a process rather than an organizational view and the Business Process Analysis approach fosters and nurtures this perspective.

How to find improvements

The information integration approach. The information integration approach utilizes eight specific analytical perspectives for evaluating the existing functional environment for change. In summary, they are:

- 1) Eliminate a step by totally automating the process. Certain steps are subject to automation. This is the easiest, and traditional, way to streamline the functionality; simply automate one or more tasks. For example, many current environments include tasks associated with loading data. In many instances these data sets can be created automatically if earlier functional activities are supported.
- 2) Change the sequence in the functional flow as a result of the computer support. The system may enable tasks to be done earlier based on data availability or reduction in the time a task takes. For example price quotations may not be based on complete manufacturing standards because of the time it takes to prepare the standards, even though it would improve the quality of the quotes. It's not productive to engineer every quote when only half the quotes actually turn into orders. Thus once it's an order it is passed through engineering for the calculation of the manufacturing standards. However, the application of information technology could make it feasible to engineer every quotation, producing better quotations and streamlining the order processing tasks since the order would already have the manufacturing standards attached when it is received.
- 3) Make steps parallel rather than sequential. In most environments modeled after a manual processing flow, the work is strung out sequentially because of the old need to pass the manual file information from one location to another. Even following the provision of computer support, the basic processes may have remained unaltered. Parallel processing based on subsets of the required data base can speed up the overall elapsed time required for processing items through the environment. Work can be directed to multiple tasks at the same time.

- 4) Group non-connected sequential steps performed by the same entity into a single step by providing needed input data earlier in the sequence. While in a given department area, all the tasks to be performed by the given group out to be performed at the same time if the data is available. For a long time it was traditional in construction companies to process the daily time sheets for payroll first in order to meet union timing requirements, and then pass the same time sheets back through the same area to do project cost accounting. A good system captures the cost accounting information at the same time the payroll is being processed, thus eliminating the redundant paper handling.
- 5) Streamline functions. It is often possible to omit steps and functions completely. The functional analysis will often uncover functional activities being performed that do not contribute to the achievement of an objective. Usually these activities were once required, but are no longer relevant. They are done because they have always been done!
- 6) Flow segmentation. Create "network" of flows based on conditional logic. In most environments, tasks cannot be subdivided by whether or not the particular item requires some action at a particular step because the person responsible for the process has to look at it to determine whether or not something needs to be done. Computer system support can perform the logic checks required and route items to a particular task area only if an action is required
- 7) Modify the performance of an individual task. It is often possible to make it easier and quicker to perform a given functional step by adding new inputs, etc. This portion of the analysis deals with the Kaizen emphasis on individual tasks. An analyst should always carefully analyze each functional task to determine whether or not the task can be performed more efficiently if new forms of support are provided. For example, if a task to determine whether or not the task can be performed more efficiently if new forms of support are provided. For example, if a task required manual verification of some factor, add the evaluation data as an input set in the system support.
- 8) Introduction of New Information Technology. Often the introduction of new technology will alter the functional flow activities in order to maximize the power and impact of the new technology.

Application of these eight perspectives will generate a multitude of ideas regarding the content and structure of the business process. In addition, the analytical process utilized provides a logical framework for the evaluation and decision making process associated with each idea. No rational proposal can be made regarding how the functionality OUGHT to be performed without understanding how it IS currently performed. We have found it very difficult to convince management to change the way in which the business functionality is being performed unless one can demonstrate what the change means in relation to the way in which the functionality is being performed today, and the VALUE to be derived by making the change.

Managing Change

The decision to integrate rather than overlay the software, while producing considerable benefit to the organization through greater user efficiency, involves certain risks. Any time one starts evaluating functional activities, one immediately becomes involved in people and their jobs. Revisions to functionality produces changes in job descriptions. Good functional analysis challenges people to critically examine their jobs, and possibly discard elements that are no longer valuable to the organization. Because of this, a strong change management process may need to be in place that anticipates and manages the fears, self-defensive postures, and organizational politics that may result from changes to task flows. Job changes and organizational structure that will be put in place along with the system must be specified as early as possible in the systems development cycle.

Defining project scope

Obviously, a key objective of the systems analysis process is the delineation of the project scope. Traditionally, scope has been difficult to establish, or constrain, as projects tend to grow and expand during the development process. Because of this phenomenon we have specified a number of analysis tasks that address the scoping issue. Scoping is not a single activity, but a progression of activities that define and refine the project scope, starting with a broad definition, but progressively tightening the definition as the requirements definition unfolds.

It starts with the project charter. In essence, the first scope statement is embodied in the problem statement developed as a part of the project charter. This "loose bounded" scope implies that project activities and resultant system development efforts will have an identifiable relationship with the problem statement.

It is further defined through context analysis. Context analysis further limits the scope by identifying the context in which the problem resides, and the context diagrams limit the system efforts to the support of the entities represented which will be a subset of the organization's total number of entities. In essence, the system scope or boundary is limited to the entities and subject matter represented, and the purpose/objectives derived from the Context Analysis.

It is further defined through task flow analysis. The next scoping activity limits the project boundary to the functional flow steps delineated as being candidates for system support. At this point the project has been further limited to the support of a specific set of business functional activities.

The identification of the output set required to support the specified functional activities within the user domain sets the scope in terms of system capability. If this scope is too large to handle within the context of a single project, then it is easier to break the project into manageable phases.

Thus by the time the requirements definition is complete, the system scope will be specifically delineated; stated in terms of the system capability to be designed, programmed, and implemented. Since benefit is determined by the impact of the output set on the business functionality, this definition of scope aligns with the determination of cost/ benefit and is ideal for the management of data processing system development activities.

Summary

While context analysis provides us with a "big picture" of process flow, it is not enough. Task flow analysis takes us to the final level of detail in Business Process Analysis, and it enables us to begin the Requirements Definition phase.

Task flow analysis involves taking processes that occur within entities on the context diagram and specifying—in the form of a flowchart—the steps that occur. It also involved looking carefully at those flowcharts to find ways to improve task flow, either through automating, infomating, or changing manual processes.
